Pegah Golchin,Jannis Weil, Ralf Kundel, Ralf Steinmetz. *Dynamic Network Intrusion Detection System in Software-Defined Networking*. Accepted for publication in Proceedings of the MaLeNe workshop (4th KuVS Fachgespraech) "Network Softwarization", 2023.

Dynamic Network Intrusion Detection System in Software-Defined Networking

Pegah Golchin *, Jannis Weil *, Ralf Kundel *, Ralf Steinmetz *
*Multimedia Communications Lab (KOM), Technical University of Darmstadt, Germany
Contact: pegah.golchin@kom.tu-darmstadt.de

***Abstract*—Software-Defined Networking (SDN) enhances network management by separating control and data plane functionalities, but the centralized control plane increases the risk of cyber attacks. Therefore, detecting network intrusions, including unknown (zero-day) attacks, is crucial. Machine learning models may be a promising solution, but often lack adaptability due to their reliance on fixed datasets during training. This study investigates corresponding challenges and outlines the potential of employing online learning methods.**

***Index Terms*—Network Intrusion Detection, SDN, Machine Learning, Online Learning, Reinforcement Learning**

I. INTRODUCTION

Software-Defined Networking (SDN) is a network architecture that separates control and forwarding functions, enhancing the network management flexibility. The centralized control plane enables a comprehensive network view by establishing forwarding rules to the switch's flow table. However, the centralized control plane is vulnerable to network intrusions [1]. For instance, a Denial of Service (DoS) attack can exhaust flow table memory, preventing the switch from accepting new legitimate flows. Moreover, an attack can overload the controller with an excessive number of new packet flows, leading to disruptions and possible network outages.

To address these issues, increasing the number of SDN controllers can distribute the load to enhance resilience and scalability [2]. However, local network intrusions can still impact the entire network, emphasizing the need for prompt detection and mitigation. Therefore, an intrusion detection system (IDS) is required for an effective network management. By learning flow patterns, Machine learning (ML) has shown promising results in the area of network flow clustering and classification. A ML-based IDS can be implemented using various techniques, including supervised [3] and unsupervised models [4]. Supervised ML-based IDS relies on labeled datasets for effective training, but its ability to detect new attacks is limited to those with similar distribution to the training dataset. Unsupervised models require no labeled data and explore data to identify patterns, leading to better detection of zero-day attacks. However, both supervised and unsupervised approaches struggle with adaptability to dynamic network architectures and concept drift over time.

Concept drift refers to changes in network architecture that result in modifications to the data distribution. Network modifications, such as adding or removing nodes, can impact routing and alter the statistical features of flows over time [5].

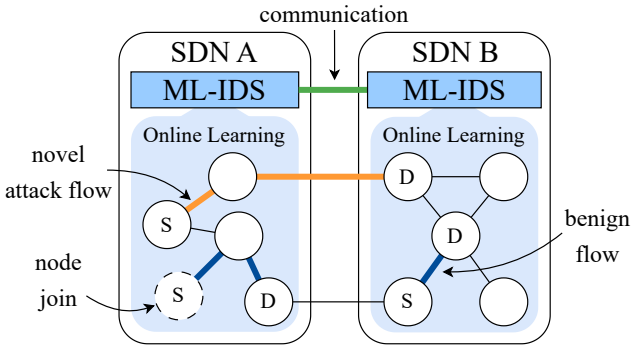

Fig. 1: A cooperative ML-based intrusion detection system (ML-IDS) refines its model in an online fashion and communicates with other instances to classify flows on a global scale.

This shift in distribution can lead to decreased performance in flow classification using fixed models. Network congestion is another example that can impact the distribution of benign flows. A static ML model may misclassify congested flows as attack flows, highlighting the problem of generalization.

To address these challenges, mechanisms are needed for model updates and adaptation to network architecture changes [6], concept drifts, and dynamic variations in flow distributions. Decentralization and cooperation among multiple SDN controllers are also crucial for detecting global trends and resolving attacks across subnetworks. The subsequent section discusses potential solutions and explores challenges associated with each approach.

II. CASE STUDY

Online learning, also known as incremental learning, can be a potential solution. It enables regular updates of a ML model as new data is obtained, allowing an IDS to continuously refine its model to adapt to concept drifts and maintain the detection performance [7]. However, online learning introduces the problem of catastrophic forgetting, where an ML-based IDS may unlearn how to handle previously seen attacks. To mitigate this, the training data should be carefully shaped to adequately represent each attack type. Consequently, the well-crafted and split dataset can be played back as a sequence to continuously refine the IDS model over time. The individual datasets have to cover distribution shift, emergence of novel attacks and reemergence of previous attacks to evaluate the applicability of the approach. As all flows are taken from

Pegah Golchin,Jannis Weil, Ralf Kundel, Ralf Steinmetz. *Dynamic Network Intrusion Detection System in Software-Defined Networking*. Accepted for publication in Proceedings of the MaLeNe workshop (4th KuVS Fachgespraech) "Network Softwarization", 2023.

available datasets, the ground truth labels for the classification problem are known. These can be compared with the predictions of the IDS model to evaluate the efficacy of the online learning mechanisms. Moreover, cooperation between individual SDN controllers in a ML-based IDS can be achieved with a distributed training paradigm like federated learning [8] to propagate long-term knowledge, or with a direct communication link between the models for reactive coordination [9]. If a neural network is used to model the IDS, hidden state information could be exchanged between IDS models to share their view on the network and improve local decision making. Communication approaches from the area of multi-agent reinforcement learning [10] could be adapted for semi-supervised online learning. The overall idea is illustrated in Fig. 1, showing multiple benign flows from source nodes (S) to destination nodes (D) within the same subnetwork and an attack flow that spans from SDN A to SDN B.

Determining the ML-based IDS refining time is another crucial metric which should be considered [11]. A naive solution entails updating the ML-based IDS deployed in the control plane whenever the data plane encounters a new unmatched flow. However, this approach can overwhelm the control plane during instances of DoS attacks or network congestion. To address this challenge, a potential solution is to update the online model using multiple trigger mechanisms.

Therefore, to detect concept drifts, two updating phases are employed. The first phase investigates a subset of flow features, denoted as $F_c = [f_i, f_j, .., f_K]$, which exhibit common behaviors seen in network intrusions (e.g., packet interarrival time, packet size). Analyzing the patterns within these features enables the determination of shifts in the underlying data distribution. When concept drift occurs, the switch initiates a process where flows are forwarded to the controller for a forwarding duration of t_f. During this time, the ML model is updated to adapt to the new data patterns. Furthermore, to address potential shifts in the data distribution caused by network architecture changes, a second trigger mechanism for updating is proposed. This approach involves selecting a random updating time interval, denoted as T_u. After this interval elapses, network flows are forwarded to the control plane for a duration of t_f to update the ML model. Selecting T_u randomly helps to prevent attackers from predicting the precise timing of the ML model updates. The proposed aggregated trigger mechanism ensures that the IDS continuously adapts and remains effective in detecting potential intrusions.

As an alternative to using existing data sets, one could simulate the underlying network and the emergence of new benign and attack flows. The interaction of the IDS model with the network could then be modelled and approached with reinforcement learning. Generative models [12] could be viable to simulate the emergence of new flows in the environment, but constraining their output to realistic traffic data will presumably require the involvement of domain experts and manually created rules. Creating new flows by manipulating existing traffic patterns would also be feasible and require less manual intervention. While this setting would be more realistic than the first approach, the expected effort for design and implementation outweighs its benefits.

III. CONCLUSION

This work explores the insufficient adaptability of existing ML-based IDS and proposes solutions for their effective use. The main challenges for leveraging online learning include catastrophic forgetting and determining model refinement time. To tackle catastrophic forgetting, we emphasize the need for well-shaped training data that represents benign and attack flows. For refinement time determination, two trigger mechanisms are proposed to detect different concept drifts. If the effectiveness of this approach can be verified in a centralized setup, a potential next step would be the extension to a decentralized setting with cooperative SDN controllers.

ACKNOWLEDGMENT

This work is funded by the Federal Ministry of Education and Research of Germany (BMBF) trough Software Campus Grant 01IS17050 (AC3Net, and ML-based NIDS), the CELTIC-NEXT Flagship Project AI-NET-PROTECT and in parts by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI.

REFERENCES

[1] P. Golchin, C. Zhou, P. Agnihotri, M. Hajizadeh, R. Kundel, and R. Steinmetz, "Cml-ids: Enhancing intrusion detection in sdn through collaborative machine learning."

[2] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103, pp. 101–118, 2018.

[3] P. Golchin, R. Kundel, T. Steuer, R. Hark, and R. Steinmetz, "Improving ddos attack detection leveraging a multi-aspect ensemble feature selection," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–5.

[4] M. Hajizadeh, S. Barua, and P. Golchin, "Fsa-ids: A flow-based self-active intrusion detection system," in *IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2023, pp. 1–9.

[5] D. C. Mulimani, S. G. Totad, and P. R. Patil, "Concept drift adaptation in intrusion detection systems using ensemble learning," *International Journal of Natural Computing Research (IJNCR)*, vol. 10, no. 4, pp. 1–22, 2021.

[6] M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, K. Rusek, S. Xiao, X. Shi, X. Cheng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Routenet-fermi: Network modeling with graph neural networks," *IEEE/ACM Transactions on Networking*, 2023.

[7] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[8] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[9] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Advances in Neural Information Processing Systems*, 2016, pp. 2244–2252.

[10] C. Zhu, M. Dastani, and S. Wang, "A survey of multi-agent reinforcement learning with communication," arXiv preprint arXiv:2203.08975 [cs.LG], 2022.

[11] L. Xie, S. Zou, Y. Xie, and V. V. Veeravalli, "Sequential (quickest) change detection: Classical results and new directions," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 2, pp. 494–514, 2021.

[12] S. Bourou, A. El Saer, T.-H. Velivassaki, A. Voulkidis, and T. Zahariadis, "A review of tabular data synthesis using gans on an ids dataset," *Information*, vol. 12, no. 09, p. 375, 2021.